



# KOREAN PATENT ABSTRACTS(KR)

Document Code:A

(11) Publication No.1020030037125 (43) Publication Date. 20030512

(21) Application No.1020010068232 (22) Application Date. 20011102

(51) IPC Code:

G06F 9/24

(71) Applicant:

ELECTRONICS AND TELECOMMUNICATIONS RESEARCH INSTITUTE

(72) Inventor:

KIM, CHAE GYU

KIM, HEUNG NAM

LIM, CHAE DEOK

PYO, CHANG U

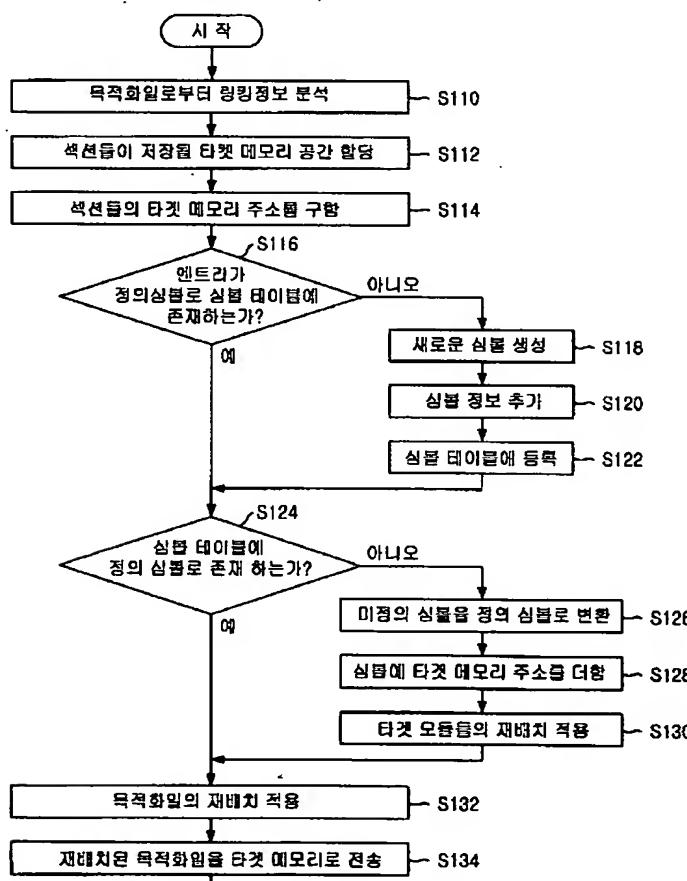
WOO, DEOK GYUN

(30) Priority:

(54) Title of Invention

GRADUAL REMOTE LOADING DEVICE FOR EMBEDDED SYSTEM AND METHOD THEREOF

Representative drawing



(57) Abstract:

PURPOSE: A gradual remote loading device for an embedded system and a method thereof are provided to execute a remote linking for linking target modules in a host and a gradual linking for loading object files one by one.

CONSTITUTION: A leader module analyzes linking information necessary for a linking for linking an object file (S110). A linker allocates a target memory space for storing sections based on section information and a section size (S112) and obtains target memory addresses of sections (S114). It is judged whether an entry exists in a symbol table as a definition symbol with respect

to each entry of symbol information(S116). It is judged whether the symbol exists in a symbol table as a definition symbol(S124). If the symbol exists in a symbol table as a definition symbol, a rearrangement of an object file is applied(S132). A text and a data section of the rearranged object file are transmitted to a target memory(S134).

© KIPO 2003

if display of image is failed, press (F5)

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)

(51) Int. Cl. <sup>7</sup> G06F 9/24	(11) 공개번호 특2003-0037125 (43) 공개일자 2003년05월12일
(21) 출원번호 10-2001-0068232 (22) 출원일자 2001년11월02일	
(71) 출원인 한국전자통신연구원 대전 유성구 가정동 161번지 임채덕	
(72) 발명자 대전광역시 유성구 전민동 엑스포아파트 306-1203 표창우 서울특별시 강남구 압구정동 현대아파트 72-901 우덕균 경기도 고양시 덕양구 토당동 삼윤아파트 1-808 김희남 대전광역시 유성구 전민동 엑스포아파트 207-1102 김채규	
(74) 대리인 대전광역시 유성구 신성동 149-13 가람빌라 2-201 특허법인 신성	

심사청구 : 있음

(54) 내장형 시스템을 위한 점진적 원격 로딩 장치 및 그 방법

**요약**

1. 청구 범위에 기재된 발명이 속한 기술 분야

본 발명은 내장형 실시간 소프트웨어를 호스트 상에서 원격 로딩해 주는 로더 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것임.

2. 발명이 해결하려고 하는 기술적 과제

본 발명은, 호스트에서 타겟 모듈들을 링킹하는 원격 링킹과 목적 화일들이 하나씩 로딩되면서 링킹이 이루어지는 점진적 링킹을 수행하도록 하는 내장형 시스템을 위한 점진적 원격 로딩 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하고자 함.

3. 발명의 해결 방법의 요지.

본 발명은, 내장형 시스템 개발 환경하의 로딩 장치에 있어서, 크로스 컴파일된 목적 화일을 프로그램 개발 도구로부터 전송받아, 목적 화일 형식에 종속되어 목적 화일을 분석하여, 목적 화일 형식에 독립적인 링킹 정보를 추출하는 목적 화일에 종속적인 리더 모듈 및 상기 리더 모듈로부터 추출된 링킹 정보를 전송받아, 추출된 링킹 정보를 이용하여 목적 화일을 상기 타겟 시스템으로 다운로딩하며, 상기 타겟 시스템의 타겟 모듈들에 대한 재배치를 수행하는 목적 화일에 독립적인 링커 모듈을 포함함.

4. 발명의 중요한 용도

본 발명은 내장형 시스템의 프로그램 개발 환경 등에 이용됨.

**대표도**

도2

**색인어**

내장형 시스템, 호스트 컴퓨터, 타겟 시스템, 원격 로딩, 점진적 로딩

**영세서**

## 도면의 간단한 설명

도 1은 일반적인 내장형 시스템 개발 환경을 보여주는 개념도.

도 2는 본 발명에 따른 점진적 원격 로딩 장치를 포함하는 타겟 관리자의 일실시에 구성도.

도 3은 본 발명에 따른 상기 도 2의 링커의 일실시에 구성도.

도 4는 본 발명에 이용되는 링킹 정보의 구성 예시도.

도 5는 본 발명의 일실시에 따른 두 C 프로그램의 링킹을 보여주는 예시도.

도 6은 본 발명의 일실시에 따른 내장형 시스템을 위한 점진적 원격 로딩 방법을 설명하기 위한 개념도.

도 7은 본 발명의 일실시에 따른 add.o 목적 화일의 점진적 원격 링킹 과정을 보여주는 설명도.

도 8은 본 발명의 일실시에 따른 내장형 시스템을 위한 점진적 원격 로딩 방법에 대한 흐름도.

\* 도면의 주요 부분에 대한 부호의 설명

210 : 전단부	220 : 로더
230 : 링커	240 : 심볼 테이블 관리자
250 : 타겟 메모리 관리자	260 : 후단부

## 발명의 상세한 설명

### 발명의 목적

#### 발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 내장형 실시간 소프트웨어를 호스트 상에서 원격 로딩해 주는 로더 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것으로, 특히 목적 화일 형식에 종속적인 모듈과 독립적인 모듈로 세분화하여 종속적인 모듈은 목적 화일로부터 화일 형식에 독립적인 링킹(linking) 정보를 추출하도록 하고, 독립적인 모듈은 이 링킹 정보로부터 실제적인 링킹을 담당하도록 하며, 이미 로딩>Loading)된 타겟 모듈들에 대해서도 재배치를 적용할 수 있도록 하는 내장형 시스템을 위한 점진적 원격 로딩 장치 및 그 방법에 관한 것이다.

정보 가전에 탑재할 내장 시스템은 작은 컴퓨터라고 볼 수 있지만, 우리가 보통 사용하는 개인용 컴퓨터와는 비교가 안될 정도로 작은 규모이다.

그래서, 디버거와 같이 리소스를 많이 필요로 하는 대형 소프트웨어를 내장 시스템에 탑재하여 응용 프로그램을 개발하기는 상당히 어려우며, 내장형 시스템은 일반적으로 자체적인 프로그램 개발이 부적절한 시스템 환경을 갖는다고 말할 수 있다.

따라서, 타겟 시스템과 시리얼 혹은 이더넷으로 연결하여 원격 컴퓨터(이하 호스트라 칭함)에서 내장 시스템에 탑재할 응용을 개발하려는 시도가 있어 왔으며, 내장형 시스템 응용 프로그램 개발 환경은 호스트 시스템이 내장형 시스템 즉, 타겟 시스템을 지원하도록 구성되었다. 호스트는 우리가 사용하는 일반적인 PC나 워크스테이션으로 타겟을 위해 프로그램 개발에 필요한 여러 작업을 지원해 주는 역할을 한다.

호스트-타겟으로 연결되는 내장형 시스템 개발 환경에서 호스트의 링커는 크로스 컴파일된 목적 화일을 타겟의 모듈들과 링킹(linking)하고 타겟으로 다운로딩한다.

호스트-타겟으로 구성되는 내장형 시스템 개발 환경은 많은 회사에서 실시간 운영체제와 함께 상품으로 제공되고 있다. 이와 같은 내장형 시스템 개발 환경에서는 호스트 시스템에서 크로스 컴파일된 목적 화일들의 타겟으로의 링킹과 로딩을 담당하는 모듈이 존재한다.

종래의 목적 화일들의 링킹과 다운로딩을 담당하는 원격 개발 환경의 링킹 모듈은 목적 화일 형식에 종속적으로 구현되었다. 이것은 새로운 목적 화일 형식에 대해서 모듈 전체가 새롭게 구성되어야 하기 때문에 목적 화일 형식이 변경되는 경우에 개발 시간이 비례하여 증가되는 문제점이 있었다.

또한, 종래의 원격 개발 환경의 링킹 모듈은 목적 화일들을 타겟으로 링킹, 다운로딩할 때, 각 화일을 상호 간에 참조 관계가 있을 때 순서에 따라 다운로딩이 되지 않으면 정확한 결과를 얻을 수 없었다. 따라서, 정확한 수행 결과를 얻기 위해서는 사용자가 반드시 목적 화일들의 다운로딩 순서를 지키거나, 호스트에서 목적 화일들을 하나로 통합하여, 통합된 목적 화일을 타겟으로 다운로딩해야 했다. 이때 사용자가 모듈들의 다운로딩 순서를 고려해야 한다는 것은 사용자에게 불편함을 주고, 여러 목적 화일들을 하나의 화일로 통합하여 다운로딩하는 것은 프로그램 개발 시간을 증가시키게 되는 문제점이 있었다.

#### 발명이 이루고자 하는 기술적 과제

따라서, 본 발명은 상기와 같은 문제점을 해결하기 위하여 제안된 것으로, 호스트에서 타겟 모듈들을 링킹하는 원격 링킹과 목적 화일들이 하나씩 로딩되면서 링킹이 이루어지는 점진적 링킹을 수행하도록 하

는 내장형 시스템을 위한 점진적 원격 로딩 장치 및 그 방법과, 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하는 것을 그 목적으로 한다.

### 발명의 구성 및 작용

상기와 같은 목적을 달성하기 위한 본 발명은, 내장형 시스템 개발 환경하의 로딩 장치에 있어서, 크로스 컴파일된 목적 화일을 프로그램 개발 도구로부터 전송받아, 목적 화일 형식에 종속되어 목적 화일을 분석하여, 목적 화일 형식에 독립적인 링킹 정보를 추출하는 목적 화일에 종속적인 리더 모듈; 및 상기 리더 모듈로부터 추출된 링킹 정보를 전송받아, 추출된 링킹 정보를 이용하여 목적 화일을 상기 타겟 시스템으로 다운로딩하며, 상기 타겟 시스템의 타겟 모듈들에 대한 재배치를 수행하는 목적 화일에 독립적인 링커 모듈을 포함하여 이루어진 것을 특징으로 한다.

그리고, 본 발명은 내장형 시스템 개발 환경하의 로딩 방법에 있어서, 목적 화일을 링킹하기 위하여 리더 모듈이 링킹에 필요한 링킹 정보를 분석하는 제 1 단계; 링커가 섹션 정보를 이용하여 섹션들이 저장될 타겟 메모리 공간을 할당하고, 섹션들의 타겟 메모리 주소를 구하는 제 2 단계; 심볼 정보의 각 엔트리들에 대하여 엔트리가 정의 심볼이고 심볼 테이블에 존재하는지를 판단하는 제 3 단계; 상기 제 3 단계의 판단 결과에 따라, 심볼이 상기 심볼 테이블에 존재하는 경우에 상기 심볼 테이블에 정의 심볼로 존재하는지를 판단하고, 존재하지 않는 경우에 상기 심볼 테이블에 새로운 심볼을 추가한 후에 정의 심볼인지를 판단하는 제 4 단계; 상기 제 4 단계의 판단 결과에 따라, 정의 심볼인 경우에 목적 화일의 재배치를 수행하고, 미정의 심볼인 경우에 정의 심볼로 전환한 후에 목적 화일의 재배치를 수행하는 제 5 단계; 및 상기 제 5 단계에서 재배치된 목적 화일을 타겟 메모리로 전송하는 제 6 단계를 포함하여 이루어진 것을 특징으로 한다.

또한, 본 발명은 프로세서를 구비한 점진적 원격 로딩 장치에, 목적 화일을 링킹하기 위하여 리더 모듈이 링킹에 필요한 링킹 정보를 분석하는 제 1 기능; 링커가 섹션 정보를 이용하여 섹션들이 저장될 타겟 메모리 공간을 할당하고, 섹션들의 타겟 메모리 주소를 구하는 제 2 기능; 심볼 정보의 각 엔트리들에 대하여 엔트리가 정의 심볼이고 심볼 테이블에 존재하는지를 판단하는 제 3 기능; 상기 제 3 기능의 판단 결과에 따라, 심볼이 상기 심볼 테이블에 존재하는 경우에 상기 심볼 테이블에 정의 심볼로 존재하는지를 판단하고, 존재하지 않는 경우에 상기 심볼 테이블에 새로운 심볼을 추가한 후에 정의 심볼인지를 판단하는 제 4 기능; 상기 제 4 기능의 판단 결과에 따라, 정의 심볼인 경우에 목적 화일의 재배치를 수행하고, 미정의 심볼인 경우에 정의 심볼로 전환한 후에 목적 화일의 재배치를 수행하는 제 5 기능; 및 상기 제 5 기능에서 재배치된 목적 화일을 타겟 메모리로 전송하는 제 6 기능을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

본 발명은 다양한 목적 화일 형식을 동적으로 지원해 주고 관련된 모듈들을 부분적으로 로딩/언로딩함으로써, 타겟 시스템과의 통신 횟수를 줄여 내장형 응용 개발 기간을 단축시키고자 한다. 즉, 본 발명은 호스트상에서 컴파일이 끝난 오브젝트 모듈이 네트워크를 통하여 타겟 보드에 로딩하는 기능을 수행함에 있어서, 오브젝트 모듈 형식에 독립적이며, 여러 오브젝트 모듈이 링크되는 응용의 일부 모듈의 로딩/언로딩을 지원하여 내장형 응용 개발 시간을 단축시키고자 한다.

이를 위해, 본 발명에서는 로더를 목적 화일 형식에 종속적인 모듈과 독립적인 모듈로 세분화하였다. 종속적인 모듈은 목적 화일로부터 화일 형식에 독립적인 링킹 정보를 추출하고, 독립적인 모듈은 이 링킹 정보로부터 실제적인 링킹을 담당한다. 이와 같은 세분화는 내장형 시스템 개발 환경에서 타겟 시스템에 대한 이식성을 높일 수 있다.

또한, 본 발명의 링커는 로딩되는 목적 화일 뿐만 아니라, 이미 로딩된 타겟 모듈에 대해서도 재배치를 적용하는 점진적 원격 링킹을 수행한다. 링커의 점진적 원격 링킹은 사용자가 목적 화일들의 링킹 순서를 고려하지 않고 임의의 순서로 링킹할 수 있는 편의성을 제공할 수 있다.

상술한 목적, 특징들 및 장점은 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이다. 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일실시예를 상세히 설명한다.

도 1은 일반적인 내장형 시스템 개발 환경을 보여주는 개념도이다.

도면을 참조하면, 내장형 시스템 개발 환경은 호스트 컴퓨터와 타겟 시스템으로 구성되며, 내장형 시스템의 응용 프로그램 개발은 내장형 시스템 보다 성능이 우수한 호스트 시스템에서 수행된다.

호스트 시스템이 목표로 하는 내장형 시스템을 타겟 시스템이라 부른다. 호스트 시스템(host system)과 타겟 시스템(target system)은 네트워크(network)를 통하여 연결되며, 호스트 시스템은 타겟 시스템을 관리하는 타겟 관리자(Target Manager)(110)와 프로그램 개발에 필요한 크로스 컴파일러(100), 디버거(104), 헬(102), 자원 모니터 등의 개발 도구들을 포함한다.

개발 도구들은 타겟 관리자(110)를 통하여 타겟 시스템과 연결된다.

타겟 시스템은 호스트의 타겟 관리자(110)와 연결되는 타겟 에이전트(120), 실시간 운영체제(122), 응용 프로그램 모듈들을 구성된다.

타겟 시스템의 독립 응용 프로그램인 타겟 에이전트(120)는 호스트의 타겟 관리자(110)와 연결하여, 호스트의 개발 도구들 또는 타겟 관리자(110)에게 프로그램 개발에 필요한 타겟 서비스를 제공한다.

도 2는 본 발명에 따른 점진적 원격 로딩 장치를 포함하는 타겟 관리자의 일실시에 구성도이다.

도면을 참조하면, 도 1의 타겟 관리자의 모듈은 프로그램 개발에 필요한 컴파일러, 헬, 디버거 등의 도구들과 연결되는 전단부(210), 타겟과 연결되는 후단부(260), 그리고 내부 모듈들(220, 230, 240, 250)로 구성된다. 내부 모듈들은 타겟 메모리를 관리하는 타겟 메모리 관리자(target memory manager)(250), 타겟에서 정의된 심볼들을 관리하는 타겟 심볼 테이블 관리자(target symbol table manager)(240), 호스

트에서 크로스 컴파일된 목적 파일을 타겟으로 다운로딩 또는 타겟에 다운로딩된 모듈들을 언로딩하는 로더(loader)(220)로 구성된다.

타겟 관리자의 로더(220)는 링커(230)를 포함하며, 링커(230)는 호스트에서 크로스 컴파일된 목적 파일을 분석하여 타겟에 다운로딩된 모듈들과의 링킹을 수행한 후 타겟에 다운로딩하는 역할을 담당한다. 링킹(linking)은 다음과 같은 단계로 진행된다.

호스트-타겟으로 연결되는 내장형 시스템 개발 환경에 포함되는 링커(230)는 호스트에서 크로스 컴파일된 목적 파일을 타겟의 모듈들과 링킹, 다운로딩하는 기능을 담당한다.

내장형 시스템 개발 환경에서의 링커(230)는 목적 파일 형식에 종속되므로, 타겟 시스템 또는 크로스 컴파일러가 변경되면 그에 따라 링커(230)도 변경되어야 한다.

이와 같은 링커(230)의 변경을 최소화하기 위하여, 본 발명에서는 링커(230)를 목적 파일 형식에 종속적인 부분과 독립적인 부분으로 구분하였다.

도 3은 본 발명에 따른 상기 도 2의 링커의 일실시예 구성도이다.

도면을 참조하면, 도 2의 링커(230)는 COFF 및 ELF 목적 파일 형식에 종속적인 리더(reader) 모듈(COFF 리더, ELF 리더)(310, 320)과 독립적인 링커 모듈(330)을 구비하고 있다.

그리고, 종속적인 리더 모듈(310, 320)은 COFF 리더(310)와 ELF 리더(320)로 구성되며, COFF 리더(310)는 COFF 목적 파일 형식에 종속적인 리더 모듈을 나타내고, ELF 리더(320)는 ELF 목적 파일 형식에 종속적인 리더 모듈을 말한다.

리더 모듈(310, 320)은 목적 파일 형식에 종속되어 목적 파일을 분석하여 목적 파일 형식에 독립적인 링킹 정보를 링커 모듈(330)에게 제공한다.

링킹 정보를 바탕으로 링커 모듈(330)은 목적 파일 형식에 독립적으로 링킹, 다운로딩을 수행한다. 여기에서 링킹 정보는 도 4와 같이 섹션 정보(410), 심볼 정보(420), 재배치 정보(430)로 구성된다.

섹션 정보(410)는 타겟 메모리 영역을 차지하는 텍스트, 데이터, bss 섹션들의 정보를 나타낸다. 섹션 타입은 텍스트, 데이터, bss 섹션들을 구분해주고, 섹션 위치는 목적 파일 내에서 해당 섹션의 옵셋 값을 말하고, 섹션 크기는 섹션의 크기 정보를 말한다. 목적 파일의 텍스트, 데이터 섹션은 섹션 위치와 크기를 바탕으로 호스트에서 타겟으로 다운로딩되고, 초기화되지 않은 데이터 값을 갖는 bss 섹션은 크기 정보를 바탕으로 타겟 메모리 영역만 할당된다.

심볼 정보(420)는 목적 파일에서 정의되는 정의 심볼과 외부에서 정의되고 목적 파일내에서 참조되는 미정의 심볼에 대한 정보를 유지한다.

심볼 타입은 정의 심볼, 미정의 심볼을 구분하여 주고, 정의 섹션은 그 심볼이 정의되는 섹션을 가리키며, 섹션 정보의 인덱스 값을 갖는다. 미정의 심볼의 경우는 0 값을 갖는다. 심볼 위치는 심볼이 정의되는 섹션내 옵셋 값을 말한다. 미정의 심볼의 경우는 0 값을 갖는다.

재배치 정보(430)는 텍스트, 데이터 섹션에 대하여 재배치를 적용할 때 사용되는 정보를 유지한다. 재배치 타입은 재배치가 적용되는 규칙, 예를 들면 재배치 계산 규칙, 재배치 적용 비트 수 등의 정보를 내포하는 정수 값을 갖는다.

재배치 적용 섹션은 재배치가 적용되는 섹션을 가리키며, 섹션 정보의 인덱스 값을 갖는다. 재배치 위치는 재배치가 적용되는 부분의 섹션내 옵셋 값을 말한다. 재배치 심볼은 재배치와 관련된 심볼을 가리키며, 심볼 정보의 인덱스 값을 갖는다.

도 5는 본 발명의 일실시예에 따른 두 C 프로그램의 링킹을 보여주는 예시도이다.

도면을 참조하면, add.c 파일에는 두 수를 더하는 add 함수와 그 결과 값이 저장되는 전역 변수 sum이 정의된다.

calc.c 파일에는 add 함수와 sum 변수를 참조하여 10과 20의 합을 구하는 calc 함수가 정의된다.

add.c로부터 크로스 컴파일된 add.o 목적 파일은 add 함수로 구성되는 텍스트 섹션과 sum 변수로 구성되는 데이터 섹션을 포함한다.

그리고, calc.c로부터 크로스 컴파일된 calc.o 목적 파일은 calc 함수로 구성되는 텍스트 섹션을 포함한다. add 함수는 add.o 목적 파일에서 정의되는 sum 변수를 참조하여, calc 함수는 add.o 파일에서 정의되는 add 함수와 sum 변수를 참조한다.

두 목적 파일의 텍스트, 데이터 섹션들은 타겟 메모리로 다운로딩되고, add 함수, sum 변수의 참조에 대한 재배치가 이루어진다.

링커는 두 목적 파일 add.o, calc.o를 링킹하기 위하여 리더 모듈은 링킹에 필요한 링킹 정보를 분석한다.

목적 파일이 COFF 형식인 경우에는 COFF 리더가 연결되어 링킹 정보가 분석되고, ELF 형식인 경우에는 ELF 리더가 연결되어 링킹 정보가 분석된다.

분석된 링킹 정보는 COFF, ELF 형식에 관계없이 단일한 형식의 정보를 포함한다. 이와 같이 분석된 링킹 정보는 도 4에서 제시된다. 각 표들의 두 번째 행은 각 링킹 정보들의 인덱스를 나타낸다.

섹션 정보에서 섹션 타입은 텍스트, 데이터, bss로 구분되고, 심볼 정보에서 심볼 타입은 정의, 미정의로 구분된다. 재배치 정보에서 재배치 타입 ABS32와 PC24는 ARM 프로세서를 목표로하는 ELF 목적 파일 형식에서 사용되는 재배치 타입의 한 예이다.

재배치 탑일 ABS32는 재배치 적용될 부분의 32비트를 심볼 주소 값으로 대체하는 것을 말하며, PC24는 재배치 적용될 부분 명령의 32비트 부분에서 24비트 부분을 심볼 주소를 PC 주소의 상대값으로 계산하여 대체하는 것을 의미한다.

여기에서 링커는 호스트에서 크로스 컴파일된 목적 파일들을 각각 하나씩 호스트에서 타겟으로 점진적으로 링킹, 다운로딩한다. 예를 들면, 도 5에서와 같이, 두 목적 파일 add.o, calc.o는 호스트에서 하나의 목적 파일로 합쳐져 타겟으로 링킹되지 않고, 각각 독립적으로 타겟으로 링킹된다. 또한, 목적 파일들의 링킹은 임의의 순서로 진행될 수 있다.

이와 같이 링킹은 원격지에 있는 모듈들과 점진적으로 진행되므로, 이와 같은 링킹 방식을 점진적 원격 링킹(incremental remote linking)이라 부른다.

도 6은 본 발명의 일실시예에 따른 내장형 시스템을 위한 점진적 원격 로딩 방법을 설명하기 위한 개념도이다.

도면에서,  $T_i$ 는 이미 타겟으로 다운로딩된  $i$ 번째 모듈, 또는 타겟으로 링킹될 호스트의  $i$ 번째 목적 파일의 텍스트 섹션을 타나낸다. 그리고,  $D_i$ 는  $i$ 번째 데이터 섹션을 말하고,  $B_i$ 는  $i$ 번째 bss 섹션을 말한다.

도면에서, 좌측 원(610)은 호스트에서 크로스 컴파일되어 타겟으로 링킹되는  $n$ 번째 목적 파일을 나타낸다.

그리고, 상단 직사각형(630)은 그 목적 파일로부터 분석된 링킹 정보를 말한다. 하단 직사각형(620)은 이미 다운로딩된 모듈들에서 정의 또는 참조되는 심볼들이 유지되는 타겟 심볼 테이블을 나타낸다. 타겟 심볼 테이블의 두 원은 임의의 심볼을 나타내고, Dsym은 정의 심볼을 말하고, Usym은 미정의 심볼을 말한다. 우측 직사각형(640)은 타겟 메모리를 나타내며, 현재  $n-1$ 개의 모듈들이 타겟 메모리에 위치하고 점선으로 표시된 직사각형은  $n$ 번째 모듈이 저장될 공간을 나타낸다. 각 원들과 직사각형들 사이의 화살표는 링킹이 진행되는 방향을 나타내고, 그 위의 숫자는 링킹이 진행되는 순서를 나타낸다.

$n$ 번째 목적 파일에 대한 타겟으로의 링킹, 다운로딩 과정은 다음과 같다.

1. 목적 파일로부터 섹션 정보, 심볼 정보, 재배치 정보등의 링킹 정보를 분석한다(도 6의 (1)).
2. 섹션 정보의 섹션 탑일과 섹션 크기를 바탕으로 텍스트 섹션  $T_n$ , 데이터 섹션  $D_n$ , bss 섹션  $B_n$ 들이 저장될 타겟 메모리 공간을 할당하며, 섹션들의 타겟 메모리 주소를 구한다(도 6의 (2)).
3. 심볼 정보의 각 엔트리들에 대하여,

[1] 엔트리가 정의 심볼이고 심볼 테이블에 존재하지 않으면(도 6의 (3)),

- a. 새로운 심볼을 생성한다.
- b. 생성된 심볼에 심볼 이름, 타겟 메모리 주소 등을 더한다.
- c. 심볼을 심볼 테이블에 등록한다.

[2] 엔트리가 정의 심볼이고 심볼 테이블에 미정의 심볼로 존재하면(도 6의 (4), (5)),

- a. 미정의 심볼을 정의 심볼로 변환한다.
- b. 심볼에 타겟 메모리 주소를 더한다.
- c. 미정의 심볼이 유지하는 재배치 정보를 이용하여, 타겟의 모듈들에 대하여 재배치를 적용한다.

4. 재배치 정보의 각 엔트리들에 대하여,

[1] 재배치와 관련된 심볼을 타겟 심볼 테이블에서 가져온다.

[2] 그 심볼이 정의 심볼이면 심볼의 타겟 메모리 주소와

[3] 엔트리의 재배치 정보를 바탕으로 호스트에서 목적 파일의 텍스트, 데이터 섹션들에 대해서 재배치를 수행한다(도 6의 (6)).

[4] 미정의 심볼이면 재배치 정보를 심볼에 더한다.

5. 재배치된 목적 파일의 텍스트, 데이터 섹션을 타겟으로 보낸다(도 6의 (7)).

도 7은 본 발명의 일실시예에 따른 add.o 목적 파일의 점진적 원격 링킹 과정을 보여주는 설명도이다.

1. add.o 목적 파일로부터 링킹 정보를 구한다(도 7의 (1)). 링킹 정보는 도 6에서 제시된다.
2. add.o 목적 파일의 섹션 정보로부터, 텍스트 섹션 56바이트와 데이터 섹션 4바이트에 대한 타겟 메모리 공간을 할당한다. 이때, 각 섹션들의 타겟 메모리 주소 200, 256을 구한다(도 7 (2)).
3. 심볼 정보의 각 엔트리 add, sum들에 대하여

[1] add 심볼에 대하여,

- a. 심볼 테이블의 미정의 심볼 add를 정의 심볼로 변환한다(도 7의 (3)).
- b. add 심볼의 타겟 메모리 주소 200과 add 심볼의 재배치 정보로부터 124번지의 add 심볼 참조 부분에 대하여 재배치를 적용한다(도 7의 (4)).

[2] sum 심볼에 대하여,

- a. 심볼 테이블의 미정의 심볼 sum을 정의 심볼로 변환한다(도 7의 (5)).
- b. sum 심볼의 타겟 메모리 주소 256과 add 심볼의 재배치 정보로부터 136번지의 sum 심볼 참조 부분에 대하여 재배치를 적용한다(도 7의 (6)).
- 4. 링킹 정보의 재배치 정보와 sum 심볼의 타겟 메모리 주소로부터 add.o 목적 파일의 sum 참조 부분에 대하여 재배치를 적용한다(도 7의 (7)).
- 5. 재배치된 add.o 목적 파일의 텍스트 섹션과 데이터 섹션을 타겟 메모리에 저장한다.

도 8은 본 발명의 일실시예에 따른 내장형 시스템을 위한 점진적 원격 로딩 방법의 흐름도이다.

도면을 참조하면, 본 발명의 일실시예에 따른 내장형 시스템을 위한 점진적 원격 로딩 방법은, 먼저 목적 파일을 링킹하기 위하여 리더 모듈은 링킹에 필요한 링킹 정보를 분석한다(S110). 이때, 목적 파일이 COFF 형식인 경우에는 COFF 리더가 연결되어 링킹 정보가 분석되고, ELF 형식인 경우에는 ELF 리더가 연결되어 링킹 정보가 분석된다. 분석된 링킹 정보는 COFF, ELF 형식에 관계없이 단일한 형식의 정보를 포함하며 섹션 정보, 심볼 정보, 재배치 정보로 구성된다.

이후, 링커는 섹션 정보의 섹션 타입과 섹션 크기를 바탕으로 섹션들이 저장될 타겟 메모리 공간을 할당하고(S112), 섹션들의 타겟 메모리 주소를 구한다(S114).

이어서, 심볼 정보의 각 엔트리들에 대하여 엔트리가 정의 심볼이고 심볼 테이블에 존재하는지를 판단한다(S116).

판단 결과, 심볼이 심볼 테이블에 존재하면 심볼 테이블에 정의 심볼로 존재하는지를 판단하고(S124). 존재하지 않으면 심볼 테이블에 새로운 심볼을 생성하고(S118) 심볼이름, 타겟 메모리 주소 등의 심볼 정보를 추가하여(S120), 심볼을 심볼 테이블에 등록한 후(S122), 심볼 테이블에 새로운 심볼을 추가한 후에 심볼 테이블에 정의 심볼로 존재하는지를 판단한다(S124).

심볼이 심볼 테이블에 정의 심볼로 존재하는지의 판단 결과, 정의 심볼로 존재하면 목적 파일의 재배치를 적용하여(S132), 정의 심볼로 존재하지 않으면 미정의 심볼을 정의 심볼로 변환하고(S126), 심볼에 타겟 메모리 주소를 더하여(S128), 미정의 심볼이 유지하는 재배치 정보를 이용하여 타겟의 모듈들에 대하여 재배치를 적용한 후에(S130), 목적 파일의 재배치를 적용한다(S132). 여기에서 목적 파일의 재배치는 재배치와 관련된 심볼을 타겟 심볼 테이블에서 가져온 후에, 그 심볼이 정의 심볼이면 심볼의 타겟 메모리 주소와 엔트리의 재배치 정보를 바탕으로 호스트에서 목적 파일의 텍스트, 데이터 섹션들에 대해서 재배치를 수행하며, 미정의 심볼이면 재배치 정보를 심볼에 더한다.

한편, 재배치된 목적 파일의 텍스트, 데이터 섹션을 타겟으로 전송함으로써(S134) 내장형 시스템을 위한 점진적 원격 로딩 과정은 종료된다.

상술한 바와 같은 본 발명의 방법은 프로그램으로 구현되어 컴퓨터로 읽을 수 있는 기록매체(씨디롬, 램, 롬, 플로피 디스크, 하드 디스크, 광자기 디스크 등)에 저장될 수 있다.

이상에서 설명한 본 발명은 전술한 실시에 및 첨부된 도면에 의해 한정되는 것이 아니고, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하다는 것이 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 있어 명백할 것이다.

### 발명의 효과

상기와 같은 본 발명에 따르면, 목적 파일들의 타겟으로의 링킹 순서에 상관 없이 타겟으로 링킹된 모듈들은 같은 실행 이미지를 유지하게 되어 응용 프로그램 개발자는 프로그램 개발 과정에서 프로그램을 구성하는 여러 목적 파일들을 타겟으로 링킹할 때, 목적 파일들의 링킹 순서를 고려하지 않아도 되므로 응용 프로그램 개발자에게 프로그램 개발에 대한 편의성을 제공할 수 있는 효과가 있다.

또한, 본 발명에 따르면, 타겟 시스템이 변경되면 링커의 독립적인 부분은 변경되지 않고, 종속적인 부분만 변경되므로 링커의 개발 시간을 단축시킬 수 있으며, 궁극적으로 전체 개발 환경의 타겟 시스템에 대한 이식성을 높일 수 있는 효과가 있다.

또한 본 발명에 따르면, 다양한 목적 파일 형식을 동적으로 지원해주고 관련된 모듈들을 부분적으로 로딩/언로딩할 수 있어서 타겟 시스템과의 통신 횟수를 줄여주게 되어 내장형 응용 개발 기간을 단축시킬 수 있는 효과가 있다.

### (57) 청구의 범위

#### 청구항 1

내장형 시스템 개발 환경하의 로딩 장치에 있어서,

크로스 컴파일된 목적 파일을 프로그램 개발 도구로부터 전송받아, 목적 파일 형식에 종속되어 목적 파일을 분석하여, 목적 파일 형식에 독립적인 링킹 정보를 추출하는 목적 파일에 종속적인 리더 모듈; 및

상기 리더 모듈로부터 추출된 링킹 정보를 전송받아, 추출된 링킹 정보를 이용하여 목적 파일을 상기 타겟 시스템으로 다운로딩하며, 상기 타겟 시스템의 타겟 모듈들에 대한 재배치를 수행하는 목적 파일에 독립적인 링커 모듈

을 포함하는 내장형 시스템을 위한 점진적 원격 로딩 장치.

## 청구항 2

제 1 항에 있어서,

상기 리더 모듈은,

크로스 컴파일된 COFF 목적 파일을 상기 프로그램 개발 도구로부터 전송받아, COFF 목적 파일 형식에 종속되어 목적 파일을 분석하여, 목적 파일 형식에 독립적인 링킹 정보를 추출하는 COFF 리더 모듈; 및

크로스 컴파일된 ELF 목적 파일을 상기 프로그램 개발 도구로부터 전송받아, ELF 목적 파일 형식에 종속되어 목적 파일을 분석하여, 목적 파일 형식에 독립적인 링킹 정보를 추출하는 ELF 리더 모듈

을 포함하는 내장형 시스템을 위한 정진적 원격 로딩 장치.

## 청구항 3

제 1 항 또는 제 2 항에 있어서,

상기 링킹 정보는,

실질적으로, 섹션 정보, 심볼 정보, 재배치 정보를 포함하는 것을 특징으로 하는 내장형 시스템을 위한 정진적 원격 로딩 장치.

## 청구항 4

제 3 항에 있어서,

상기 링커 모듈은,

로딩되는 목적 파일 뿐만 아니라, 이미 로딩된 타겟 모듈들에 대해서도 재배치를 적용되는 정진적 원격 링킹을 수행하되, 정진적 원격 링킹은 사용자가 목적 파일들의 링킹 순서를 고려하지 않고 임의의 순서로 링킹할 수 있는 편의성을 제공하는 것을 특징으로 하는 내장형 시스템을 위한 정진적 원격 로딩 장치.

## 청구항 5

내장형 시스템 개발 환경하의 로딩 방법에 있어서,

목적 파일을 링킹하기 위하여 리더 모듈이 링킹에 필요한 링킹 정보를 분석하는 제 1 단계;

링커가 섹션 정보를 이용하여 섹션들이 저장될 타겟 메모리 공간을 할당하고, 섹션들의 타겟 메모리 주소를 구하는 제 2 단계;

심볼 정보의 각 엔트리들에 대하여 엔트리가 정의 심볼이고 심볼 테이블에 존재하는지를 판단하는 제 3 단계;

상기 제 3 단계의 판단 결과에 따라, 심볼이 상기 심볼 테이블에 존재하는 경우에 상기 심볼 테이블에 정의 심볼로 존재하는지를 판단하고, 존재하지 않는 경우에 상기 심볼 테이블에 새로운 심볼을 추가한 후에 정의 심볼인지를 판단하는 제 4 단계;

상기 제 4 단계의 판단 결과에 따라, 정의 심볼인 경우에 목적 파일의 재배치를 수행하고, 미정의 심볼인 경우에 정의 심볼로 전환한 후에 목적 파일의 재배치를 수행하는 제 5 단계; 및

상기 제 5 단계에서 재배치된 목적 파일을 타겟 메모리로 전송하는 제 6 단계

를 포함하는 내장형 시스템을 위한 정진적 원격 로딩 방법.

## 청구항 6

제 5 항에 있어서,

상기 제 1 단계는,

크로스 컴파일러된 목적 파일의 파일 형식을 판단하는 제 7 단계;

상기 제 7 단계의 판단 결과, 목적 파일이 COFF 형식인 경우에, COFF 리더가 연결되어 링킹 정보를 분석하는 제 8 단계; 및

상기 제 8 단계의 판단 결과, 목적 파일이 ELF 형식인 경우에, ELF 리더가 연결되어 링킹 정보를 분석하는 제 9 단계

를 포함하는 내장형 시스템을 위한 정진적 원격 로딩 방법.

**청구항 7**

제 5 항 또는 제 6 항에 있어서,  
 상기 제 4 단계의 새로운 심볼을 추가하는 과정은,  
 엔트리가 정의심볼로 상기 심볼 테이블에 존재하지 않으면 새로운 심볼을 생성하는 제 10 단계;  
 심볼이름, 타겟 메모리 주소를 포함한 심볼 정보를 생성된 심볼에 추가하는 제 11 단계; 및  
 심볼 정보가 추가된 생성된 심볼을 상기 심볼 테이블에 등록하여 상기 심볼 테이블에 새로운 심볼을 추가하는 제 12 단계  
 를 포함하는 내장형 시스템을 위한 점진적 원격 로딩 방법.

**청구항 8**

제 5 항 또는 제 6 항에 있어서,  
 상기 제 5 단계의 미정의 심볼을 정의 심볼로 전환하는 과정은,  
 심볼이 상기 심볼 테이블에 정의 심볼로 존재하는지의 판단 결과, 미정의 심볼로 존재하면, 미정의 심볼을 정의 심볼로 변환하는 제 10 단계; 및  
 심볼에 타겟 메모리 주소를 더하고, 미정의 심볼이 유지하는 재배치 정보를 이용하여 타겟의 모듈들에 대해 재배치를 적용하는 제 11 단계  
 를 포함하는 내장형 시스템을 위한 점진적 원격 로딩 방법.

**청구항 9**

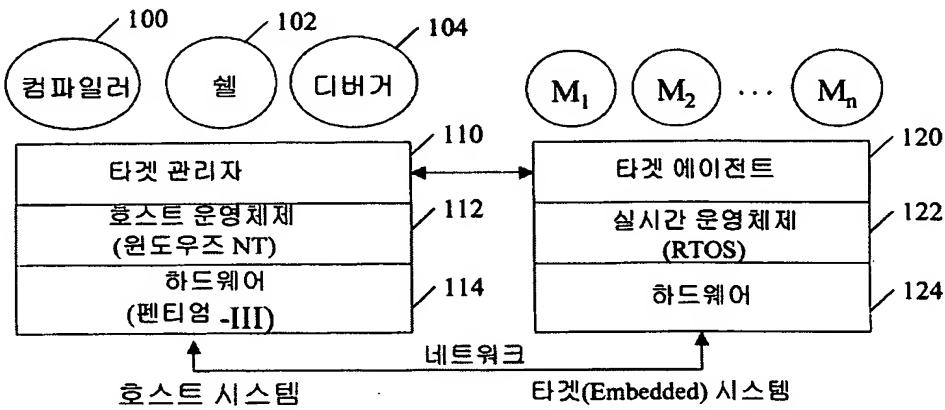
제 5 항 또는 제 6 항에 있어서,  
 상기 제 5 단계의 목적 파일을 재배치하는 과정은,  
 재배치와 관련된 심볼을 상기 심볼 테이블에서 가져오는 제 10 단계;  
 상기 제 10 단계에서 가져온 심볼이 정의 심볼이면, 심볼의 타겟 메모리 주소와 엔트리의 재배치 정보를 바탕으로 호스트에서 목적 파일의 텍스트, 데이터 섹션들에 대해서 재배치를 수행하는 제 11 단계; 및  
 상기 제 10 단계에서 가져온 심볼이 미정의 심볼이면, 재배치 정보를 심볼에 더하는 제 12 단계  
 를 포함하는 내장형 시스템을 위한 점진적 원격 로딩 방법.

**청구항 10**

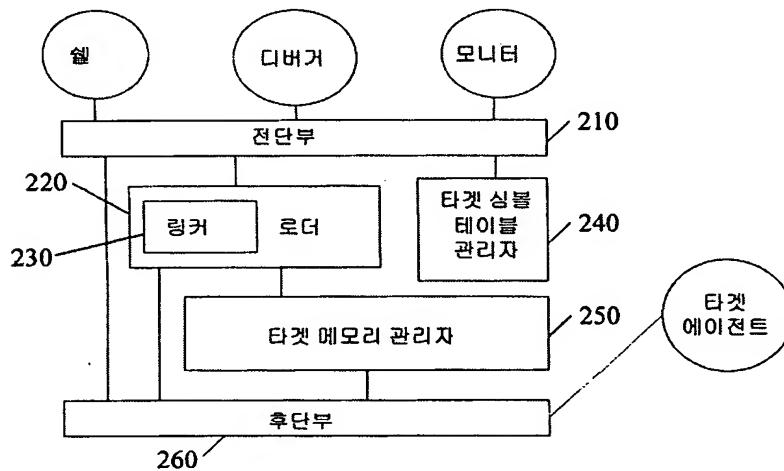
프로세서를 구비한 점진적 원격 로딩 장치에,  
 목적 파일을 링킹하기 위하여 리더 모듈이 링킹에 필요한 링킹 정보를 분석하는 제 1 기능;  
 링커가 섹션 정보를 이용하여 섹션들이 저장될 타겟 메모리 공간을 할당하고, 섹션들의 타겟 메모리 주소를 구하는 제 2 기능;  
 심볼 정보의 각 엔트리들에 대하여 엔트리가 정의 심볼이고 심볼 테이블에 존재하는지를 판단하는 제 3 기능;  
 상기 제 3 기능의 판단 결과에 따라, 심볼이 상기 심볼 테이블에 존재하는 경우에 상기 심볼 테이블에 정의 심볼로 존재하는지를 판단하고, 존재하지 않는 경우에 상기 심볼 테이블에 새로운 심볼을 추가한 후에 정의 심볼인지를 판단하는 제 4 기능;  
 상기 제 4 기능의 판단 결과에 따라, 정의 심볼인 경우에 목적 파일의 재배치를 수행하고, 미정의 심볼인 경우에 정의 심볼로 전환한 후에 목적 파일의 재배치를 수행하는 제 5 기능; 및  
 상기 제 5 기능에서 재배치된 목적 파일을 타겟 메모리로 전송하는 제 6 기능  
 을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

**도면**

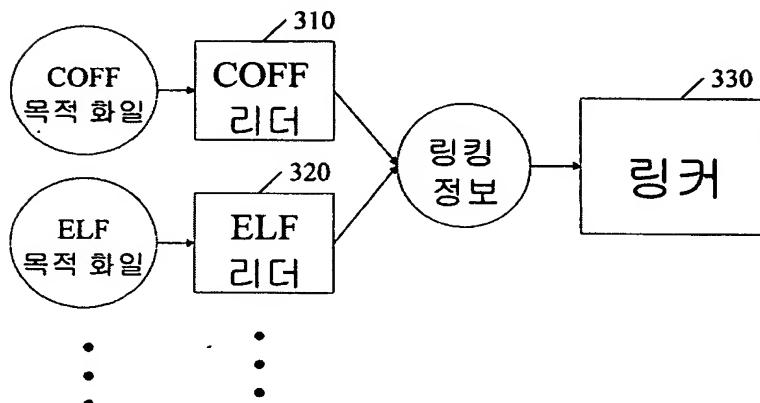
도면1



도면2



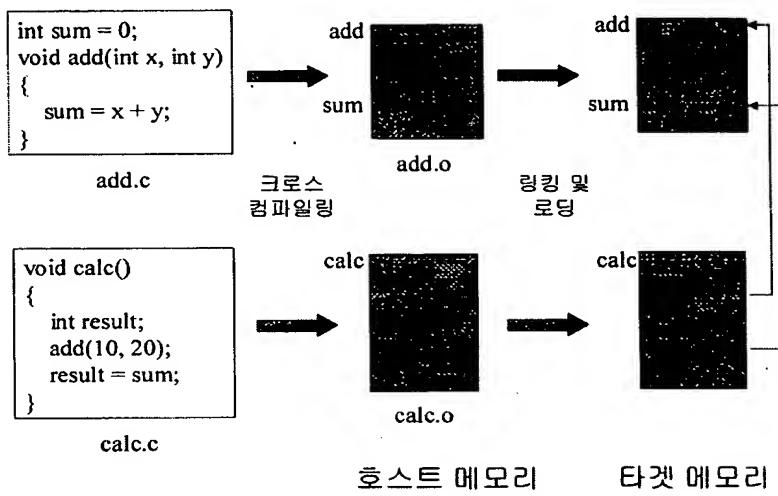
도면3



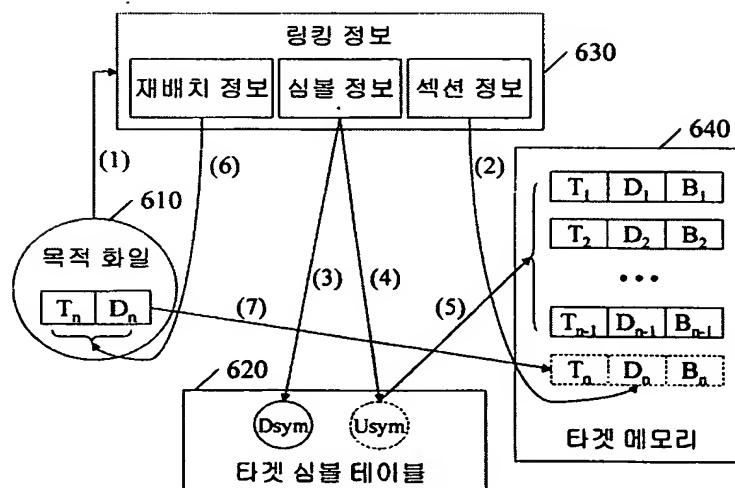
도면4

410	420	430
섹션 정보	심볼 정보	재배치 정보
섹션 타입	심볼 타입	재배치 타입
섹션 위치	심볼 이름	재배치 적용 섹션
섹션 크기	정의 섹션	재배치 위치
	심볼 위치	재배치 심볼

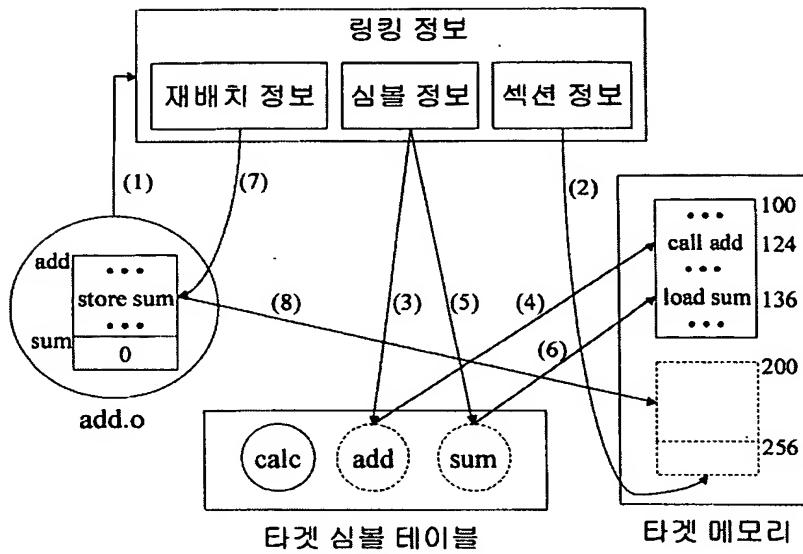
도면5



도면6



## 도면7



## 도면8

